

**Amendments to the Claims:**

This listing of claims will replace all prior versions, and listings, of claims in the application:

**Listing of Claims:**

**Claims 1-2 (cancelled).**

**Claim 3 (currently amended):** The tool of claim 44 **[[2]]**, wherein the framework identifies a priority for each of the test modules.

**Claim 4 (original):** The tool of claim 3, wherein the control module is operable to cause the test modules to be executed sequentially according to the priority identified in the framework for each of the test modules.

**Claim 5 (currently amended):** The tool of claim 44 **[[1]]**, wherein a mechanism is provided for identifying the at least one test module as being one of active and not active.

**Claim 6 (original):** The tool of claim 5, wherein the mechanism for identifying the at least one test module as being one of active and not active is included in the framework.

**Claim 7 (original):** The tool of claim 5, wherein the mechanism for identifying the at least one test module as being one of active and not active is included in the control module.

**Claim 8 (currently amended):** The tool of claim 44 **[[2]]**, wherein the framework comprises a test directory having a plurality of entries, each entry identifying one of the plurality of test modules.

**Claim 9 (original):** The tool of claim 8, wherein entry defines a priority for the one of the test modules identified therein.

**Claim 10 (original).** The tool of claim 8, wherein the identity of the one of the test modules defines its priority.

Claims 11-13 (cancelled).

**Claim 14 (currently amended):** A program storage device readable by a computer, tangibly embodying a program of instructions executable by the computer to perform method steps for verifying the correct installation of a software installation package without performing a test execution of a software component of the software installation package that includes at least one installed software component and wherein the software package includes a file list having data entries associated with have information about parameters for the at least one software component, the method comprising the steps of:

a) providing a plurality of test modules configured to compare the data entries from the file list with information about an associated software component of the software package to determine correct installation of the software component;

b) a) providing a framework for identifying at least one the test modules, each said test module defining a test that, when executed, compares the data entries from the file list with associated information about an associated installed software component of the software package; of at least one parameter of the at least one software component of the package wherein said test is configured to use the data entries of the file list to test the at least one parameter of the software package and wherein the testing is accomplished without performing a test execution of a software component of the software installation package;

c) b) accessing the framework to identify the plurality of test modules; at least one test module; and

d) e) causing a first the at least one test module to compare selected data entries from the file list with associated information about an installed software component of the software package to verify the correctness of the installation of the software component; perform the test defined thereby on the package;

e) determining whether further test modules of the plurality of test modules are to be executed;

f) where further test modules are to be executed, causing another test module to compare selected data entries from the file list with associated information about another installed software component of the software package to verify the correctness of the installation of the another software component, then returning to operation e);

g) where no further test modules are to be executed, ending the program.

Claim 15 (cancelled).

**Claim 16 (currently amended):** The method of claim 14 ~~[[15]]~~, wherein a priority for each of the test modules is identified in the framework.

**Claim 17 (currently amended):** The method of claim 14 ~~[[15]]~~, comprising sequentially causing each of the test modules to be executed according to the priority identified for each of the test modules.

**Claim 18 (currently amended):** The method of claim 14 ~~[[15]]~~, comprising identifying each of the test modules as being one of active and not active and wherein only active test modules are executed.

**Claim 19 (currently amended):** The method of claim 14 ~~[[15]]~~, comprising providing a directory in the framework, wherein the directory has a plurality of entries, each entry identifying one of the plurality of test modules.

**Claims 20-22 (cancelled).**

**Claim 23 (currently amended):** A method for verifying the correct installation of a software ~~installation~~ package that includes at least one installed software component ~~wherein verifying is conducted without performing a test execution of the software components of the software installation package~~ and wherein the software package includes a file list installed with the software package, the file list having data entries associated with parameters for the at least one software component, the method comprising the operations ~~[[steps]]~~ of:

a) providing a plurality of test modules configured to compare the data entries from the file list with information about an associated software component of the installed software package to determine correct installation of the software component;

b) a) providing a framework for identifying at least one a plurality of test modules, each said test module configured to compare file data of a selected one of the installed software components to associated use the data entries of the installed file list to test the correctness of the at least one parameter of the software package thereby defining a test of at least one parameter of the at least one software component of the package;

c) b) accessing the framework to identify the plurality of test modules; at least one test module; and

d) e) causing a first ~~the at least one~~ test module to compare selected data entries from the file list with associated information about an installed software component of the software package to verify the correctness of the installation of the software component; perform the test defined thereby on the package.

e) determining whether further test modules of the plurality of test modules are to be executed;

f) where further modules are to be executed, causing another test module to compare selected data entries from the file list with associated information about another installed software component of the software package to verify the correctness of the installation of the another software component, then returning to operation e);

g) where no further modules are to be executed, ending the program.

Claim 24 (cancelled).

Claim 25 (currently amended): The method of Claim 23 ~~[[24]]~~, wherein a priority for each of the test modules is identified in the framework.

Claim 26 (original): The method of claim 25, comprising sequentially causing each of the test modules to be executed according to the priority identified for each of the test modules.

Claim 27 (currently amended): The method of Claim 23 ~~[[24]]~~, comprising identifying each of the test modules as being one of active and not active.

Claim 28 (currently amended): The method of Claim 23 ~~[[24]]~~, comprising providing a directory in the framework, wherein the directory has a plurality of entries, each entry identifying one of the test modules.

Claim 29 (currently amended): The method of Claim 23 ~~[[24]]~~, wherein each entry defines a priority of the test module identified thereby.

Claim 30 (original): The method of claim 28, wherein identity of a module defines its priority.

Claim 31 (currently amended): A method of verifying a software package that includes at least one software component, the method comprising the steps of:

a) providing a computer having installed thereon a receiving the software package comprising at least one software component wherein the software package includes an associated [[a]] file list, the file list having data entries corresponding to actual contents of the files of parameters for the at least one software component;

b) accessing a framework mounted on the computer that references at least one test module to identify the at least one test module from the framework, each said test module configured to compare selected data entries from the file list with the actual contents of the files of a software component associated with the data entries to verify the correctness of the installation of the software component; ~~use the data entries of the file list to define a test of the software package;~~ and

c) performing the test ~~defined by~~ causing the at least one test module to compare selected data entries from the file list with the actual contents of the files of a software component verify the correctness of the installation of the software component. on the package wherein testing is conducted without performing a test execution of the software components of the software installation package.

**Claim 32 (previously presented):** The method of Claim 31, including repeating steps (b) and (c) to perform a sequence of tests, the order in which the tests are performed being determined by relative priorities assigned to each of the at least one test module.

**Claim 33 (currently amended):** A computer readable medium having stored thereon a data structure operable for use in verifying a software package that includes at least one software component, the data structure comprising:

a) a first field containing data representing one of a plurality of test modules, each test module being operable to test of at least one parameter of the at least one software component of the package,

b) where data representing ones of the test modules may be added to and deleted from the data structure, creating a flexible data structure.

**Claim 34 (original):** The medium according to claim 33, wherein the data structure further comprises a second field identifying a priority for each of the test modules represented by the data in the first data field, the priority defining an order of execution of test modules.

**Claim 35 (original):** The medium according to claim 33, wherein the data structure further comprises a third field identifying the one of a plurality of test modules represented by the data in the first data field as being one of active and not active.

**Claim 36-37 (cancelled).**

**Claim 38 (currently amended):** A method of verifying the correct installation of a ~~[[that]]~~ software ~~installation~~ package that includes at least one software component, the method comprising the steps of:

a) providing a computer having installed thereon a SOLARIS © compliant software package wherein the package includes having at least one software component and an associated file list containing data about the files of the software package;

b) a) providing a test framework that includes a control module and at least one test module suitable for referencing at least one test module for conducting verification of software installation packages, the test modules configured to test the software component by comparing data from the file list with actual data from installed files of the software component ;

c) receiving a command to verify installation of the software package;

d) authenticating the command to verify;

b) receiving the software installation package;

e) e) executing an initial verification of the software installation package using a control module, wherein the initial verification is conducted by comparing data from the file list with actual data from installed files of the software component without performing a test execution of the software components of the software installation package;

f) d) where the initial verification is successful, selecting a next module from among the at least one test module to conduct verification of the software package; and

g) e) executing verification testing of the software package using the next module wherein the verification using the next module is conducted by comparing data from the file list with actual data from installed files of the software component without performing a test execution of the software components of the software installation package.

**Claim 39 (currently amended):** The method of Claim 38 wherein operations f and g ~~d and e~~ are iteratively performed until all test modules of the at least one test module have been executed.

**Claim 40 (currently amended):** The method of Claim 38 wherein if any verification or authentication operations fail an error message is generated.

**Claim 41 (currently amended):** The method of Claim 38 wherein executing the authentication of the command to verify installation ~~initial verification of the software installation package~~ comprises:

receiving a verification command; and  
checking that the verification command includes a correct number of arguments.

**Claim 42 (currently amended):** The method of Claim 41 wherein executing the authentication of the command to verify installation ~~initial verification of the software installation package~~ comprises at least one of:

confirming that a user of the software installation package has the correct permissions;  
confirming that software components of the software installation package exist in the correct directories;  
confirming that the software components do not include any zero size files;  
confirming that the software components comprise actual data files and not data links;  
and  
confirming that package map and package information files for the software package exist in a SOLARIS © ~~Solaris~~ environment.

**Claim 43 (currently amended):** The method of Claim 38 wherein b) ~~a)~~ providing a test framework that includes a control module and at least one test module ~~suitable for conducting verification of software installation packages~~ includes adding, deleting, and modifying the at least one test module to provide a flexible test framework.

**Claim 44 (currently amended):** ~~The tool of claim 2,~~ An automated software package verification tool mounted on a computer for verifying a software package that has been installed on the computer, the software package including at least one software component

and a file list having data entries that have information about the at least one software component, the tool comprising:

a plurality of test modules configured to use the data entries of the file list to verify correct installation of the software components of the software package by comparing the data entries from the file list with information about an associated software component of the software package;

a framework operable to identify the test modules wherein the plurality of test modules of the framework define a series of automatically executable tests ~~and wherein the test modules do not require the test execution of software components of the software installation package;~~

and

a control module operable to access the framework to cause the at least one test module identified therein to perform a test comparing the data entries from the file list with information about an associated software component of the software package thereby verifying the package.

**Claim 45 (cancelled).**

**Claim 46 (currently amended):** The tool of claim 44, wherein the software package is a SOLARIS® compliant software package; and

wherein the file list that identifies the plurality of software components in the software package comprises a Solaris pkgmap file.

**Claim 47 (cancelled).**

**Claim 48 (currently amended):** The tool of claim 46 [[47]], wherein the data included in the file list includes parameter information concerning the plurality of software components in the software installation package; and

wherein the plurality of test modules are operable to use the parameter information included in the file list to verify the software package.

**Claim 49 (previously presented):** The tool of claim 48, wherein the parameter information in the file list includes at least one of: file names for the software components, version numbers for the software components, vendor identification for the software components, copyright



information concerning the software components, the size of the software components, the binary data types of the software components.

**Claim 50 (previously presented):** The tool of claim 48, wherein the parameter information in the file list includes the compiler version used with the software components.

**Claim 51 (cancelled).**

**Claim 52 (currently amended):** The tool of Claim 44 [[2]] wherein test modules of the framework can be added, deleted, or modified creating a flexible framework.

**Claim 53 (cancelled).**

**Claim 54 (currently amended):** The method of claim 14 [[15]], wherein the file list that identifies the plurality of software components in the software package comprises a Solaris pkgmap file.

**Claim 55 (cancelled).**

**Claim 56 (currently amended):** The method of claim 54 [[55]], wherein comparing selected data entries from the file list with associated information about an installed software component of the software package to verify the correctness of the installation of the software component includes comparing the data from the file list with actual file information from the installed software components to determine correctness of at least one of:

~~the parameter information in the file list includes at least one of:~~ file names for the software components, version numbers for the software components, vendor identification for the software components, copyright information concerning the software components, the size of the software components, the binary data types of the software components.

**Claim 57 (currently amended):** The method of claim 56 [[55]], wherein comparing selected data entries from the file list with associated information about an installed software component of the software package further comprises comparing the parameter

~~information in the file list includes~~ the compiler version used with the installed software components with the compiler version in the file list.

**Claim 58 (currently amended):** The method of claim 14 ~~[[55]]~~, wherein comparing selected data entries from the file list with associated information about an installed software component of the software package further comprises comparing the parameter ~~information in the file list includes~~ at least one of: copyright information concerning the software installation components, the size of the software components, the binary data types of the software components.

**Claim 59 (currently amended):** The method of claim 14 ~~[[55]]~~, wherein the providing the framework includes added and deleted test modules from the framework as desired.

**Claims 60-62 (cancelled).**

**Claim 63 (previously presented):** The method of claim 23, wherein the plurality of test modules are configured to test at least one of: that the software components are compiled using the same compiler version; that binary data types of the software components are compatible with the same architecture; that the copyrights are current; and a test whether only specified software components are present in the installation package.

**Claim 64 (previously presented):** The method of claim 63, wherein the plurality of test modules are further configured to include a module for testing whether there have been changes to the software installation package relative to a prior version of the software installation package.

**Claim 65 (previously presented):** The method of claim 63, wherein the plurality of test modules are further configured to include a module for testing whether there are any zero size files in the software installation package.